UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/614,664	07/07/2003	Dikran Meliksetian	POU920020113US1	7571

7590 08/14/2006

John E. Campbell
IBM Corporation
2455 South Road, P386
Poughkeepsie, NY 12601

EXAMINER	
LE, MICHAEL	

ART UNIT	PAPER NUMBER
2163	

DATE MAILED: 08/14/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/614,664	MELIKSETIAN ET AL.	
	Examiner	Art Unit	
	Michael Le	2163	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 May 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *26 May 2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Summary and Status of Claims

1. This Office Action is in response to Applicant's reply filed May 26, 2006.

2. Claims 1-30 are pending.

3. Claims 1-4, 8, 10, 13, 14, 17, 19, 20 and 27-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dodds et al. (US Patent Pub 2002/0116371) of record, in view of Fernandez et al. (US Patent 6,604,100) of record.

4. Claims 5-7, 9, 11, 12, 15, 16, 18, 22, 23, 25 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dodds et al. (US Patent Pub 2002/0116371) of record, in view of Fernandez et al. (US Patent 6,604,100) of record, further in view of Fox et al. (US Patent Pub 2003/0120665) of record.

5. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Dodds et al. (US Patent Pub 2002/0116371) of record, in view of Fernandez et al. (US Patent 6,604,100) of record, further in view of Kerwin (US Patent Pub 2003/0212660) of record.

6. Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over Dodds et al. (US Patent Pub 2002/0116371) of record, in view of Fernandez et al. (US Patent 6,604,100) of record, further in view of Fox et al. (US Patent Pub 2003/0120665) of record, further in view of Kerwin (US Patent Pub 2003/0212660) of record.

7. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

Claim Rejections - 35 USC § 101

8. **Claims 29 and 30 are rejected under 35 U.S.C. 101** because the claimed invention is directed to non-statutory subject matter.

9. The basis of this rejection is set forth in a test of whether the invention is categorized as a process, machine, manufacture or composition of matter and if the invention produces a useful, concrete and tangible result. Mere ideas in the abstract (i.e., abstract idea, law of nature, natural phenomena) are found to be non-statutory subject matter. For a method claim to pass muster, the recited process must produce a useful, concrete and tangible result.

10. In the present case, **claims 29 and 30** recite a system, however the components of the system are merely software per se. A system claim must recite physical structure thus enabling it to be properly categorized in one of the statutory categories of invention. Since the components of the systems claims in claims 29 and 30 are software per se and do not contain any physical components, the systems can not be categorized in one of the statutory categories of invention and is thus nonstatutory.

11. To expedite a complete examination of the instant application, the claims rejected under 35 U.S.C. 101 (nonstatutory) above are further rejected as set forth below in anticipation of applicant amending these claims to place them within the four statutory categories of invention.

Claim Rejections - 35 USC § 103

12. Claims 1-4, 10, 13, 14, 17, 19, 20 and 27-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dodds et al. (US Patent Pub 2002/0116371) of record, hereinafter

"Dodds", in view of Fernandez et al. (US Patent 6,604,100) of record, hereinafter "Fernandez".

13. In regards to **claims 1**, Dodds discloses a computer program product for creating an XML representation of data stored in a relational database system (Dodds: para. 0007), the computer program product comprising a computer readable medium having computer readable program code therein (Dodds: para. 0020), the computer program product comprising:

a. computer readable program code for specifying a set of conditions that data to be retrieved from the relational database system must satisfy (Dodds: para. 0041, lines 3-5)[1];

b. computer readable program code for specifying an output format that the XML representation must satisfy (Dodds: para. 0041, lines 7-11, 13-18)[2];

c. computer readable program code for retrieving data from the relational database using a standard database access method (Dodds: para. 0042-0045)[3]; and

14. Dodds does not expressly disclose creating from the set of conditions and the format, a mapping description being in a markup language comprising XSL and SQL function, said mapping description including a user defined template identifying a defined procedure for retrieving data from the relational database, retrieving data using the mapping description and formatting the XML object representing the retrieved data using the mapping description. Dodds does disclose outputting an XML object representing the retrieved data (Dodds: para. 0041, lines

[1] A user query is interpreted as a set of conditions. It is a query to a relational database, so it is specifying conditions for data to be retrieved from the relational database system.
[2] The XML is reconstructed according to the sections (format) that are retrieved, which are specified by the user's query.
[3] SQL is used (standard database access method) to retrieve the data from the database.

7-11, 13-18). Dodds further discloses in the background information that XSL is used to format XML data to be displayed (Dodds: para. 0003, lines 11-14).

15. Fernandez discloses a method for converting relational data into a structured document (Fernandez: col. 2, lines 27-29). Fernandez further discloses an RXL language for generating a query to retrieve relational data and transform it into an XML view (Fernandez: col. 5, lines 4-8). Fernandez also discloses composing an executable query, which is comprised of a data-extraction part of one or more SQL queries and an XML-construction part, such as an XML template (Fernandez: col. 5, lines 26-30). The data-extraction part is then used to retrieve data from the relational database and the XML-construction part produces the XML document, which is then returned to the user (Fernandez: col. 5, lines 38-45).

16. Dodds and Fernandez are analogous art because they are from the same field of endeavor of mapping relational data to XML.

17. At the time of the invention it would have been obvious to a person of ordinary skill in the art to modify the computer program product of Dodds by adding computer readable code for creating from the set of conditions and the format, a mapping description (execution query taught by Fernandez) being in a markup language comprising SQL and XSL function (disclosed by Dodds in the background), retrieving data using the mapping description and formatting the XML object representing the retrieved data using the mapping description, as taught by Fernandez.

18. The motivation for doing so would have been because it is desirable to devise an efficient means of storing, indexing and retrieving via queries XML documents in a relational database

system (Dodds: para. 0004, lines 1-4). Fernandez discloses an efficient tool for querying relational data in XML, which meets the needs of Dodds (Fernandez: col. 2, lines 39-33).

19. In regards to **claim 13**, Dodds discloses updating information stored in a relational database based on data extracted from an XML file (Dodds: para. 0024, lines 1-4). Dodds also discloses retrieving data from a relational database and outputting the data as XML as addressed in the rejection to claim 1 above. Claim 13 seems to be the opposite of claim 1. Also addressed above, Fernandez discloses composing an executable query, which is comprised of a data-extraction part of one or more SQL queries and an XML-construction part, such as an XML template (Fernandez: col. 5, lines 26-30). The data-extraction part is then used to retrieve data from the relational database and the XML-construction part produces the XML document, which is then returned to the user (Fernandez: col. 5, lines 38-45). The executable query is interpreted as the mapping description. Accordingly, claim 13 is rejected because it contains the same elements as claim 1 and at the time of the invention, it would have been obvious to one of ordinary skill in the art to do the opposite of claim 1, to store the XML documents in a relational database. The motivation for doing so would have been because it is desirable to retrieve data that is stored.

20. In regards to **claim 2**, the limitation was addressed above in the rejection to claim 1. Fernandez also discloses composing an executable query, which is comprised of a data-extraction part of one or more SQL queries (first section) and an XML-construction part, such as an XML template (second section) (Fernandez: col. 5, lines 26-30).

21. In regards to **claim 3**, the limitation was addressed above in the rejection to claim 1. Fernandez also discloses composing an executable query, which is comprised of a data-extraction part of one or more SQL queries (first section) and an XML-construction part, such as an XML template (second section) (Fernandez: col. 5, lines 26-30). One or more SQL queries is interpreted as a list of individual search conditions, wherein each condition is a SQL query.

22. In regards to **claim 4**, the limitation was addressed above in the rejection to claim 1. Fernandez also discloses composing an executable query, which is comprised of a data-extraction part of one or more SQL queries (first section) and an XML-construction part, such as an XML template (second section) (Fernandez: col. 5, lines 26-30). The SQL queries in the execution query specify one or more tables upon which the statement will act (Fernandez: col. 9, line 27; col. 10, line 9), one or more columns of the table that will be retrieved (Fernandez: col. 10, line 8), a list of parameters for determining a location where a value of a parameter is to be obtained (Fernandez: col. 9, lines 28-29; col. 10, lines 10-12) and the value of the parameter to be used in a condition clause of the SQL prepared statement (Fernandez: col. 9, lines 28-29; col. 10, lines 10-12).

23. In regards to **claim 10**, Dodds discloses in the background information that XSL is commonly used for formatting XML documents (Dodds: para. 0003, lines 11-14).

24. Dodds does not expressly discloses an internal representation of the data retrieved from the database according to the specification that is formatted to be output.

25. Fernandez discloses an internal representation of the data retrieved from the database according to the specification that is formatted to be output (Fernandez: col. 5, lines 38-45).

26. At the time of the invention it would have been obvious to one of ordinary skill in the art to modify the computer program product of Dodds by adding computer readable code for applying XSL syntax to transform an internal representation of the data retrieved from the database according to the specification that is formatted to be output, taught by Fernandez.

27. The motivation for doing so would have been because it is desirable to devise an efficient means of storing, indexing and retrieving via queries XML documents in a relational database system (Dodds: para. 0004, lines 1-4). Fernandez discloses an efficient tool for querying relational data in XML, which meets the needs of Dodds (Fernandez: col. 2, lines 39-33).

28. In regards to **claim 14**, the mapping description was addressed above in the rejection to claim 13 as being disclosed by Fernandez. Fernandez discloses composing an executable query, which is comprised of a data-extraction part of one or more SQL queries and an XML-construction part, such as an XML template (Fernandez: col. 5, lines 26-30). The data-extraction part is then used to retrieve data from the relational database and the XML-construction part produces the XML document, which is then returned to the user (Fernandez: col. 5, lines 38-45). The executable query is interpreted as the mapping description. At the time of the invention, it would have been obvious to one of ordinary skill in the art to modify the XML-construction part to be an XML retrieval part for extracting data from the XML file. The motivation for doing so would have been because it is desirable to retrieve data that is stored.

29. In regards to **claim 17**, the limitation was addressed above in the rejection to claim 13. Fernandez discloses composing an executable query, which is comprised of a data-extraction

part of one or more SQL queries and an XML-construction part, such as an XML template (Fernandez: col. 5, lines 26-30). The data-extraction part is then used to retrieve data from the relational database and the XML-construction part produces the XML document, which is then returned to the user (Fernandez: col. 5, lines 38-45). The executable query is interpreted as the mapping description. The data-extraction part contains one or more SQL queries (sequence of database operations and their respective inputs).

30. In regards to **claim 19**, Dodds discloses database operations for storing (insert) XML data in a relational database (Dodds: para. 0036, lines 7-8).

31. In regards to **claim 20**, Dodds discloses database operations that are SQL prepared statements (Dodds: para. 0043).

32. **Claim 27** is substantially similar to claim 1 in the form of a method and is rejected for the same reasons.

33. **Claim 28** is substantially similar to claim 13 in the form of a method and is rejected for the same reasons.

34. **Claim 29** is substantially similar to claim 1 in the form of a system and is rejected for the same reasons.

35. **Claim 30** is substantially similar to claim 13 in the form of a system and is rejected for the same reasons.

36. **Claims 5-9, 11, 12, 15, 16, 18, 22, 23, 25 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dodds et al. (US Patent Pub 2002/0116371) of record, hereinafter "Dodds", in view of Fernandez et al. (US Patent 6,604,100) of record, hereinafter "Fernandez", further in view of Fox et al. (US Patent Pub 2003/0120665) of record, hereinafter "Fox".**

37. In regards to **claim 5**, Dodds and Fernandez do not expressly disclose creating a first runtime object representing elements of the mapping description wherein the run-time object represents a search condition as a JDBC prepared statement and defines Java variables for the values of the parameters.

38. Fox discloses a method of transforming data from one data schema to another data schema (Fox: para. 0103), wherein the source and target schemas are analyzed and mapped, then transformed based on the mappings (Fox: para. 0107). Fox further discloses that the method creates a Java program that executes SQL query using the JDBC library (run-time object using JDBC prepared statement having Java variables for values of the parameters) (Fox: para. 0127, lines 3-7). Fox also discloses XSLT transformations wherein an XSLT script is derived to perform the transformation from a source schema to a target schema (Fox: para. 0149, lines 7-11) wherein the XSLT script is executed by a Java XSLT engine and deployed to an enterprise application infrastructure product which uses the Xalan XSLT engine to run the XSLT scripts, which are optimized as Java classfiles (Fox: para. 0152-0153).

39. Dodds, Fernandez and Fox are analogous art because they are all directed towards the same field of endeavor of data transformation.

40. At the time of the invention, it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds and Fernandez by adding computer readable code for creating a first runtime object representing elements of the mapping description wherein the run-time object represents a search condition as a JDBC prepared statement and defines Java variables for the values of the parameters, as taught by Fox.

41. The motivation for doing so would have been because XML is a standard format for data exchange between inter-enterprise applications (Fernandez: col. 1, lines 20-23) and Java is known for its advantage of portability. Using Java to run-time objects and accessing the databases with JDBC (Java Database Connectivity) is in line with the purpose of XML, which is to maintain portability when exchanging data between inter-enterprise applications.

42. In regards to **claim 6**, Dodds and Fernandez disclose producing an internal representation of the data retrieved from the database (Fernandez: col. 5, lines 4-8), providing a value for a variable needed for a first SQL prepared statement (Fernandez: col. 6, lines 29-34), executing the first SQL prepared statement (Fernandez: col. 5, lines 4-8) and storing a result of the executing step as an internal representation (Fernandez: col. 5, lines 8-10).

43. Dodds and Fernandez do not expressly disclose that the first run-time object produces the internal representation.

44. Fox discloses a method of transforming data from one data schema to another data schema (Fox: para. 0103), wherein the source and target schemas are analyzed and mapped, then transformed based on the mappings (Fox: para. 0107). Fox further discloses that the method creates a Java program that executes SQL query using the JDBC library (run-time object using

JDBC prepared statement having Java variables for values of the parameters) (Fox: para. 0127, lines 3-7).

45. At the time of the invention, it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds and Fernandez by adding computer readable code for the first run-time object to produce the internal representation, as taught by Fox.

46. The motivation for doing so would have been because XML is a standard format for data exchange between inter-enterprise applications (Fernandez: col. 1, lines 20-23) and Java is known for its advantage of portability. Using Java to run-time objects and accessing the databases with JDBC (Java Database Connectivity) is in line with the purpose of XML, which is to maintain portability when exchanging data between inter-enterprise applications.

47. In regards to **claim 7**, Dodds and Fernandez do not expressly disclose invoking an XSL transformation engine and operating on the internal representation with the first run-time object.

48. Fox discloses that the method creates a Java program that executes SQL query using the JDBC library (run-time object using JDBC prepared statement having Java variables for values of the parameters) (Fox: para. 0127, lines 3-7). Fox also discloses XSLT transformations wherein an XSLT script is derived to perform the transformation from a source schema to a target schema (Fox: para. 0149, lines 7-11) wherein the XSLT script is executed by a Java XSLT engine and deployed to an enterprise application infrastructure product which uses the Xalan XSLT engine to run the XSLT scripts, which are optimized as Java classfiles (Fox: para. 0152-0153).

49. At the time of the invention, it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds and Fernandez by adding computer readable code for invoking an XSL transformation engine and operating on the internal representation with the first run-time object, taught by Fox.

50. The motivation for doing so would have been because XML is a standard format for data exchange between inter-enterprise applications (Fernandez: col. 1, lines 20-23), XSL is used for formatting XML (Dodds: para. 0003, lines 11-14) and Java is known for its advantage of portability. Using Java to run-time objects and accessing the databases with JDBC (Java Database Connectivity) is in line with the purpose of XML, which is to maintain portability when exchanging data between inter-enterprise applications.

51. In regards to **claim 8**, Dodds discloses generating software events (Dodds: para. 0024, line 6)[4] and generating data associated with the software events (Dodds: para. 0024, lines 7-9)[5].

52. Dodds does not expressly disclose sequencing the software events and the associated data based on the internal representation of the retrieved data.

53. Fernandez discloses a method for converting relational data into a structured document (Fernandez: col. 2, lines 27-29). Fernandez further discloses an RXL language for generating a query to retrieve relational data and transform it into an XML view (internal representation) (Fernandez: col. 5, lines 4-10).

54. At the time of the invention it would have been obvious to one of ordinary skill in the art to modify the computer program product of Dodds by adding computer readable program code

[4] The database processes the query (the database processing the query is a software event).

for sequencing the software events and the associated data based on the internal representation of the retrieved data, taught by Fernandez.

55. The motivation for doing so would have been because it is desirable to devise an efficient means of storing, indexing and retrieving via queries XML documents in a relational database system (Dodds: para. 0004, lines 1-4). Fernandez discloses an efficient tool for querying relational data in XML, which meets the needs of Dodds (Fernandez: col. 2, lines 39-33).

56. In regards to **claim 9**, Dodds and Fernandez do not expressly disclose providing the software events and associated data to an XSL transformation, the providing being performed for a single event. Dodds does disclose in the background information that XSL is used for formatting XML (Dodds: para. 0003, lines 11-14).

57. Fox discloses XSLT transformations wherein an XSLT script (software events and associated data) is derived to perform the transformation from a source schema to a target schema (Fox: para. 0149, lines 7-11) wherein the XSLT script is executed (single event) by a Java XSLT engine and deployed to an enterprise application infrastructure product which uses the Xalan XSLT engine to run the XSLT scripts, which are optimized as Java classfiles (Fox: para. 0152-0153).

58. At the time of the invention it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds and Fernandez by adding computer readable code for providing the software events and associated data to an XSL transformation, the providing being performed for a single event, taught by Fox.

---

[5] The query results are generated (data associated with the processing (software event)).

59. The motivation for doing so would have been because XML is a standard format for data exchange between inter-enterprise applications (Fernandez: col. 1, lines 20-23), XSL is used for formatting XML (Dodds: para. 0003, lines 11-14) and Java is known for its advantage of portability. Using Java to run-time objects and accessing the databases with JDBC (Java Database Connectivity) is in line with the purpose of XML, which is to maintain portability when exchanging data between inter-enterprise applications.

60. In regards to **claim 11**, Dodds and Fernandez do not expressly disclose creating a second run-time object using a Xalan Templates object for representing the XSL transformation.

61. Fox discloses XSLT transformations wherein an XSLT script is derived to perform the transformation from a source schema to a target schema (Fox: para. 0149, lines 7-11) wherein the XSLT script is executed by a Java XSLT engine and deployed to an enterprise application infrastructure product which uses the Xalan XSLT engine to run the XSLT scripts, which are optimized as Java classfiles (Fox: para. 0152-0153).

62. At the time of the invention it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds and Fernandez by adding computer readable code for creating a second run-time object using a Xalan Templates object for representing the XSL transformation, taught by Fox.

63. The motivation for doing so would have been because XML is a standard format for data exchange between inter-enterprise applications (Fernandez: col. 1, lines 20-23), XSL is used for formatting XML (Dodds: para. 0003, lines 11-14) and Java is known for its advantage of portability. Using Java to run-time objects and accessing the databases with JDBC (Java

Database Connectivity) is in line with the purpose of XML, which is to maintain portability when exchanging data between inter-enterprise applications. Furthermore, Xalan is known XSL Transformation processor which is commonly used with Java as it is implemented as a Java library.

64. In regards to **claim 12**, Dodds and Fernandez do not expressly disclose creating a run-time object, saving the run-time object in a run-time object cache and using the cached run-time object.

65. Fox discloses that the method creates a Java program that executes SQL query using the JDBC library (run-time object using JDBC prepared statement having Java variables for values of the parameters) (Fox: para. 0127, lines 3-7). Fox further discloses run-time transformations (run-time objects) that are created (Fox: para. 0164, lines 1-2), saved to a cache (run-time cache) (Fox: para. 0164, lines 3-8) and used from the cache (Fox: para. 0164, lines 7-8).

66. At the time of the invention, it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds and Fernandez by adding computer readable code for creating a run-time object, saving the run-time object in a run-time object cache and using the cached run-time object, as taught by Fox.

67. The motivation for doing so would have been because it saves times because the object need not be generated again (Fox: para. 0166, lines 10-16).

68. In regards to **claim 15**, Dodds discloses retrieving XML object data (Dodds: para. 0024, lines 1-4). Fernandez discloses creating an internal representation of retrieved data (Fernandez: col. 5, lines 4-10).

69. Dodds and Fernandez do not expressly disclose specifying in XSL syntax a location of the XML object data to be retrieved from the XML object, wherein the XSL syntax determines a transformation and creating the internal representation of required data from the XML object data based on the transformation.

70. Fox discloses a method of transforming data from one data schema to another data schema (Fox: para. 0103), wherein the source and target schemas are analyzed and mapped, then transformed based on the mappings (Fox: para. 0107). Fox further discloses that the method creates a Java program that executes SQL query using the JDBC library (run-time object using JDBC prepared statement having Java variables for values of the parameters) (Fox: para. 0127, lines 3-7). Fox also discloses XSLT transformations wherein an XSLT script is derived to perform the transformation from a source schema to a target schema (Fox: para. 0149, lines 7-11) wherein the XSLT script is executed by a Java XSLT engine and deployed to an enterprise application infrastructure product which uses the Xalan XSLT engine to run the XSLT scripts, which are optimized as Java classfiles (Fox: para. 0152-0153).

71. At the time of the invention, it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds and Fernandez by adding computer readable program code for specifying in XSL syntax a location of the XML object data to be retrieved from the XML object, wherein the XSL syntax determines a transformation and

creating the internal representation, taught by Fernandez, of required data from the XML object data based on the transformation, taught by Fox.

72. The motivation for doing so would have been because XML is a standard format for data exchange between inter-enterprise applications (Fernandez: col. 1, lines 20-23), XSL is used for formatting XML (Dodds: para. 0003, lines 11-14) and Java is known for its advantage of portability. Using Java to run-time objects and accessing the databases with JDBC (Java Database Connectivity) is in line with the purpose of XML, which is to maintain portability when exchanging data between inter-enterprise applications.

73. In regards to **claims 16, 18, 22 and 25**, the limitations were addressed above in the rejection to claims 13 and 17 as being disclosed by Fox. Fox discloses that the method creates a Java program that executes SQL query using the JDBC library (run-time object using JDBC prepared statement having Java variables for values of the parameters) (Fox: para. 0127, lines 3-7). Fox also discloses XSLT transformations wherein an XSLT script is derived to perform the transformation from a source schema to a target schema (Fox: para. 0149, lines 7-11) wherein the XSLT script is executed by a Java XSLT engine and deployed to an enterprise application infrastructure product which uses the Xalan XSLT engine to run the XSLT scripts, which are optimized as Java classfiles (Fox: para. 0152-0153). Fox discloses using Xalan (claim 16), XSL scripts (claim 18), Java run-time objects using JDBC prepared statements and Java variables (claim 22) and the process of creating the run-time object and running an XSL script using an XSL processing engine (claim 25).

74. In regards to **claim 23**, Dodds and Fernandez disclose associating with an SQL statement, a value of a variable from the internal representation of retrieved data and executing the SQL statement (Fernandez: col. 5, lines 13-21).

75. **Claim 26** is essentially claim 12 and is rejected for the same reasons.

76. **Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Dodds et al. (US Patent Pub 2002/0116371) of record, hereinafter "Dodds", in view of Fernandez et al. (US Patent 6,604,100) of record, hereinafter "Fernandez", further in view of Kerwin (US Patent Pub 2003/0212660) of record.**

77. Dodds and Fernandez do not expressly disclose sequencing database operations as segmented transactional groups.

78. Kerwin discloses a transaction server that provides sequenced SQL statements (Kerwin: para. 0075). Kerwin further discloses queued SQL statements (transactional groups) (Kerwin: para. 0179).

79. Dodds, Fernandez and Kerwin are analogous art because they are all directed towards the same field of endeavor of data management using databases and XML.

80. At the time of the invention, it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds and Fernandez by adding computer readable program code for sequencing database operations as segmented transactional groups, taught by Kerwin.

81. The motivation for doing so would have been because it is desirable to ensure that commands occur in a proper sequence thereby giving an expected result (Kerwin: para. 0166).

82. **Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over Dodds et al. (US Patent Pub 2002/0116371) of record, hereinafter "Dodds", in view of Fernandez et al. (US Patent 6,604,100) of record, hereinafter "Fernandez", further in view of Fox et al. (US Patent Pub 2003/0120665) of record, hereinafter "Fox", further in view of Kerwin (US Patent Pub 2003/0212660) of record.**

83. Dodds, Fernandez and Fox do not expressly disclose database operations that are performed by respecting predefined transactional boundaries such that an operation in a group is only completed if all the operations of the group can be executed successfully.

84. Kerwin discloses a transaction server that provides sequenced SQL statements (Kerwin: para. 0075). Kerwin further discloses queued SQL statements (transactional groups) (Kerwin: para. 0179). Kerwin also discloses that the next statement in a queue is not executed until receiving confirmation of successful execution of the currently executed statement (Kerwin: para. 0175).

85. Dodds, Fernandez, Fox and Kerwin are analogous art because they are all directed towards the same field of endeavor of data management using databases and XML.

86. At the time of the invention, it would have been obvious to one of ordinary skill in the art to modify the combined computer program product of Dodds, Fernandez and Fox by adding computer readable program code for performing database operations by respecting predefined transactional boundaries such that an operation in a group is only completed if all the operations of the group can be executed successfully, taught by Kerwin.

87. The motivation for doing so would have been because it is desirable to ensure that commands occur in a proper sequence thereby giving an expected result (Kerwin: para. 0166).

Response to Amendment

Drawings

88. Applicant's amendment to the drawings to address missing reference numbers in the Specification and drawings is acknowledged. Consequently, objection to the drawings is withdrawn.

Specification

89. Applicant's amendment to the Specification to correct informalities is acknowledged. Consequently, objection to the specification is withdrawn.

Objection to claims 16 and 22 for Minor Informalities

90. Applicant's amendment to claims 16 and 22 to address the minor informalities is acknowledged. Consequently, the objection to claims 16 and 22 is withdrawn.

Rejection of Claims 7-9 and 19 under 35 U.S.C 112, Second Paragraph

91. Applicant's amendment to claims 7-9 and 19 is acknowledged. The rejection to claims 7-9 and 19 under 35 U.S.C. 112, second paragraph is withdrawn.

Response to Arguments

Rejection of Claims 29 and 30 under 35 U.S.C 101

92. Applicant's arguments in regards to the rejection of claims 29 and 30 under 35 U.S.C. 101 have been fully considered but they are not persuasive. Applicant argues that the mapper recited in claims 29 and 30 is, in the preferred embodiment of the invention, a processor and thus provides physical structure to the system. However, nowhere in the Specification is a mapper explicitly defined as a processor. In other words, a mapper being a processor is strictly one interpretation and the other being that a mapper is merely a software module as interpreted by the Examiner. Given the possibility of a software module interpretation, claims 29 and 30 potentially do not meet the statutory requirement under 35 U.S.C. 101. Consequently, the rejection to claims 29 and 30 under 35 U.S.C. 101 is maintained.

Rejection of claims 1-30 under 35 U.S.C. 103(a)

93. Applicant's arguments in regards to the rejections to claims 1-30 under 35 U.S.C. 103(a), have been fully considered but they are not persuasive. Applicant alleges that the combination of Dodds et al. (US Patent Pub 2002/0116371) hereinafter "Dodds", and Fernandez et al. (US Patent 6,604,100), hereinafter "Fernandez", fails to disclose the newly amended limitation "said mapping description including a user defined template identifying a defined procedure for retrieving data from the relational database" (Pages 22-23 of the Remarks). The Examiner respectfully disagrees.

94. Fernandez discloses a method for converting relational data into a structured document (Fernandez: col. 2, lines 27-29). Fernandez further discloses an RXL language for generating a

query to retrieve relational data and transform it into an XML view (Fernandez: col. 5, lines 4-8). Fernandez further discloses an administrator (user) writes (defines) a view query that defines the XML virtual view of the database. The view query is then used to determine how the relational data is retrieved and formed into XML (Fernandez: col. 5, lines 4-8). The view query is then used for composing an executable query, which is comprised of a data-extraction part of one or more SQL queries and an XML-construction part, such as an XML template (Fernandez: col. 5, lines 26-30). The data-extraction part is then used to retrieve data from the relational database and the XML-construction part produces the XML document, which is then returned to the user (Fernandez: col. 5, lines 38-45). Fernandez also discloses that an administrator may choose to execute the XML view (retrieve data from the relational database) (Fernandez: col. 5, lines 46-51). Thus, Fernandez clearly discloses including a user defined template identifying a defined procedure for retrieving data from the relational database.

95. Consequently, the rejection to claims 1-30 under 35 U.S.C. 103(a) is maintained.

Conclusion

96. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

97. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

98. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael Le whose telephone number is 571-272-7970. The examiner can normally be reached on Mon-Thurs : 9:30am-6pm, Fri: 8am-4:30pm.

99. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Don Wong can be reached on 571-272-1834. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DON WONG
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Michael Le
Art Unit 2163
August 7, 2006